

**IN THE UNITED STATES PATENT AND TRADEMARK OFFICE**

In re application of: **Jamison**

Serial No.: **10/761,991**

Filed: **January 21, 2004**

**For: Method for Determining a Close  
Approximate Benefit of Reducing  
Memory Footprint of a Java  
Application**

§  
§  
§  
§  
§  
§

Group Art Unit: **2193**

Examiner: **Vu, Tuan A.**

Attorney Docket No.: **RSW920030277US1**

**DRAFT CLAIMS FOR ENABLING EXAMINER AMENDMENT**

## DRAFT CLAIMS FOR ENABLING EXAMINER AMENDMENT

Amendments to the Claims:

This listing of claims will replace all prior versions, and listings, of claims in the application:

Listing of Claims:

1. (Currently Amended) A computer-implemented method of improving performance in a Java computer application program executable by a Java virtual machine (JVM), comprising the steps of:  
 obtaining information associated with garbage collection; and  
 deducing changes in performance that will result from modifying the Java computer application program;

wherein a cost of garbage collection to program performance of a modified version of the Java computer application program is estimated by the computer-implemented method using a duration of an average garbage collection event and a frequency of occurrence of particular garbage collection events[.];

wherein the duration depends on an amount of garbage that must be cleaned up[.]; an algorithm used to do collecting or copying[.]; a heap compaction[.]; a cost of reconciling object references that are moved[.]; and a number of Java-based finalizers that must be executed[.and];

wherein the frequency depends on a rate of object creation[.]; heap fragmentation[.]; size of the heap[.]; and a garbage collection policy;

wherein the Java computer application program is changed by reducing memory from a footprint of the Java computer application program, wherein given an amount of memory to be reduced from the footprint ( $m$ ), a total duration for a run ( $d$ ), and how much of the run is spent in garbage collection ( $g$ ), a number of additional transactions that can be computed during the run is determined according to:

$y = \text{function}(m, t, g, d, f)$ , where

$t$  is current computed throughput of the Java computer application program in transactions/second,  
 $f$  is average memory footprint of the Java computer application program during duration  $d$ , and  
 $y$  is the number of additional transactions gained [[after]] as a result of reducing the memory footprint of the Java computer application program to  $(f - m)$  during the total duration  $d$ .

2-7. (Cancelled)

## DRAFT CLAIMS FOR ENABLING EXAMINER AMENDMENT

8. (Previously Presented) The method of claim 1, wherein the information associated with garbage collection is obtained from a verbosegc.

9. (Cancelled)

10. (Currently Amended) A computer system capable of running a Java application program by a Java virtual machine (JVM), comprising a computer and further comprising:

a garbage heap associated with garbage collection events, wherein garbage collection events have an average duration and frequency; and

instructions for estimating changes in performance resulting from modification of the Java application program using information obtained about the garbage collection events;

wherein a cost of garbage collection to program performance of a modified version of the Java application program is estimated by the computer system using a duration of an average garbage collection event and a frequency of occurrence of particular garbage collection events[.];

wherein the duration depends on an amount of garbage that must be cleaned up[.]; an algorithm used to do collecting or copying[.]; a heap compaction[.]; a cost of reconciling object references that are moved[.]; and a number of Java-based finalizers that must be executed[. and];

wherein the frequency depends on a rate of object creation[.]; heap fragmentation[.]; size of the heap[.]; and a garbage collection policy;

wherein the Java application program is changed by deducting memory from a footprint of the Java application program, wherein given an amount of memory to be deducted from the footprint ( $m$ ), a total duration for a run ( $d$ ), and how much of the run is spent in garbage collection ( $g$ ), a number of additional transactions that can be computed during the run is determined according to:

$$y = \text{function}(m, t, g, d, f), \text{ where}$$

$t$  is current computed throughput of the Java computer application program in transactions/second,

$f$  is average memory footprint of the Java computer application program during duration  $d$ , and

$y$  is the number of additional transactions gained [[after]] as a result of reducing the memory footprint of the Java computer application program to  $(f - m)$  during the total duration  $d$ .

11-15. (Cancelled)

## DRAFT CLAIMS FOR ENABLING EXAMINER AMENDMENT

16. (Previously Presented) The method of claim 10, wherein the information associated with garbage collection is obtained from a verbosegc.

17. (Currently Amended) A computer program product stored in a computer readable medium for improving performance in a Java computer application program executable by a Java virtual machine (JVM), comprising:

first instructions for obtaining information associated with garbage collection; and

second instructions for deducing changes in performance that will result from modifying the Java computer application program, wherein a cost of garbage collection to program performance of the Java computer application program is estimated by the computer program product using a duration of an average garbage collection event and a frequency of occurrence of particular garbage collection events[.];

wherein the duration depends on an amount of garbage that must be cleaned up[.]; an algorithm used to do collecting or copying, a heap compaction[.]; a cost of reconciling object references that are moved[.]; and a number of Java-based finalizers that must be executed[. and];

wherein the frequency depends on a rate of object creation[.]; heap fragmentation[.]; size of the heap[.]; and a garbage collection policy;

wherein the Java computer program is changed by deducting memory from a footprint of the Java computer application program, wherein given an amount of memory to be deducted from the footprint ( $m$ ), a total duration for a run ( $d$ ), and how much of the run is spent in garbage collection ( $g$ ), a number of additional transactions that can be computed during the run is determined according to:

$$y = \text{function } (m, t, g, d, f), \text{ where}$$

$t$  is current computed throughput of the Java computer application program in transactions/second,

$f$  is average memory footprint of the Java computer application program during duration  $d$ , and

$y$  is the number of additional transactions gained [[after]] as a result of reducing the memory footprint of the Java computer application program to  $(f - m)$  during the total duration  $d$ .

18-22. (Cancelled)

23. (Previously Presented) The method of claim 17, wherein the information associated with garbage collection is obtained from a verbosegc.